# Backend-Business-Logik – vollständiger Entwurf (v1)

Inhalt

[Backend-Business-Logik – vollständiger Entwurf (v1) 1](#_Toc210578363)

[1. Klassen 1](#_Toc210578364)

[2. Zustände & Übergänge (State Machines) 4](#_Toc210578365)

[2.1 Flugzeug 4](#_Toc210578366)

[2.2 Reservierung 4](#_Toc210578367)

[2.3 Pilotenflug 4](#_Toc210578368)

[2.4 Start-/Landeliste (Movement) 5](#_Toc210578369)

[3. Validierungen 5](#_Toc210578370)

[4. Berechnungen 5](#_Toc210578371)

[4.1 Blockminuten 5](#_Toc210578372)

[4.2 Flugkosten 5](#_Toc210578373)

[4.3 Landegebühr 5](#_Toc210578374)

[4.4 Steuern & Beträge 5](#_Toc210578375)

[5. Rollen 6](#_Toc210578376)

[6. Fehlercodes (Beispiele) 6](#_Toc210578377)

[7. Konfigurierbarkeit 6](#_Toc210578378)

[8. Offene Punkte (optional, falls später gebraucht) 6](#_Toc210578379)

## 1. Klassen

**t\_kostenstelle**

* Id
* Name
* Strasse
* PLZ
* Ort
* Aktiv

**t\_benutzer**

* Id
* Vorname
* Nachname
* Email
* Passwort
* KostenstelleId → *t\_kostenstelle.Id*

**t\_rolle**

* Id
* Rollentyp

**t\_benutzerrolle**

* BenutzerId → *t\_benutzer.Id*
* RolleId → *t\_rolle.Id*
* **Primärschlüssel:** (BenutzerId, RolleId)

**t\_preiskategorie**

* Id
* Kategorie
* Preis

**t\_flugart**

* Id
* Flugtyp

**t\_flugzeugtyp**

* Id
* Flugzeugtyp
* PreisId → *t\_preiskategorie.Id*

**t\_flugzeug**

* Id
* Kennzeichen
* Zaehlerstand (DECIMAL(6,1))
* Grounded (BOOLEAN)
* FlugzeugtypId → *t\_flugzeugtyp.Id*
* KostenstelleId → *t\_kostenstelle.Id*
* DatumARC
* DatumFlugzeitwartung
* Wartungsart

**t\_flugminutenkosten**

* Id
* FlugartId → *t\_flugart.Id*
* FlugzeugId → *t\_flugzeug.Id*
* KostenMinuten
* **Unique:** (FlugartId, FlugzeugId)

**Vererbungsstruktur**

**t\_flug *(Basistabelle)***

* Id
* FlugzeugId → *t\_flugzeug.Id*
* FlugartId → *t\_flugart.Id*
* StartTs
* EndeTs
* Bemerkung
* Status ∈ {Geplant, StartErfasst, EndeErfasst, Storniert}

**t\_reservierung *(is-a t\_flug)***

* Id → *t\_flug.Id*
* Pilot (VARCHAR)
* Datum
* Gastname
* Gastkontakt
* Status ∈ {Offen, Erledigt, Storniert}

**t\_pilotenflug *(is-a t\_flug)***

* Id → *t\_flug.Id*
* Pilot (VARCHAR)
* StartZaehler (DECIMAL(6,1))
* EndZaehler (DECIMAL(6,1))
* DauerMinuten (berechnet)

**t\_startlandeliste**

* Id
* Einflug (BOOLEAN)
* Ausflug (BOOLEAN)
* Startzeit
* Landezeit
* Bemerkung
* Bezahlt (BOOLEAN)
* Pilot1
* Pilot2
* FlugzeugId → *t\_flugzeug.Id*
* FlugartId → *t\_flugart.Id*

## 2. Zustände & Übergänge (State Machines)

### 2.1 Flugzeug

* **Zustände:** Aktiv, Gegroundet
* **Übergang Gegroundet→Aktiv:** nur Rolle {Wart, Admin} ändern
* **Nebenwirkungen:** Reservieren/Starten blockieren

### 2.2 Reservierung

* **Zustände:** Offen → {Erledigt | Storniert}
* **Regeln:**
  + Kollisionen verboten
  + Flugzeug muss Aktiv sein zur Anlage
  + Storno: Slot freigeben, referenzierte Flüge trennen

### 2.3 Pilotenflug

* **Zustände:** StartErfasst → EndeErfasst
* **Start:** Zählerstand == letzter Zählerstand (sonst Fehler; Override nur {Wart, Admin})
* **Ende:** Zählerstand := eingabe

### 2.4 Start-/Landeliste (Movement)

* **Zustände pro Zeile:** Offen → Bezahlt
* **Regeln:**
  + Startzeit ≥ Landezeit (falls Start vorhanden)
  + EF/AF konsistent (EF=Einflug? AF=Abflug?)
  + Checkliste erfüllt
  + Landegebühr berechnen, Aktionen: Neu/Kopieren/Schlepp/Touch&Go, Rechnungen erzeugen

## 3. Validierungen

* **Zeiten:** endzeit > startzeit; Reservierung innerhalb Tagesgrenzen

## 4. Berechnungen

### 4.1 Blockminuten

blockmin = Startzähler − Endzähler

### 4.2 Flugkosten

preis\_minute = in Flugzeug oder in der Preiskategorie

flugkosten = blockmin \* preis\_minute

### 4.3 Landegebühr

landung\_basis = flugzeug preis

### 4.4 Steuern & Beträge

netto = sum(positionen)  
steuer = netto \* steuersatz  
brutto = netto + steuer

## 5. Rollen

| Aktion | Normal/Schüler | Wart | Abrechnung | Admin |
| --- | --- | --- | --- | --- |
| Login, eigene Reservierungen/Flüge ansehen | ✔ | ✔ | ✔ | ✔ |
| Reservierung anlegen/ändern (aktiv. Flzg) | ✔ | ✔ | ✔ | ✔ |
| Zähler-Override bei Startfehler | ✖ | ✔ | ✖ | ✔ |
| Flugzeug Grounding ändern | ✖ | ✔ | ✖ | ✔ |
| Preismodell/Preis-Override setzen | ✖ | ✔ | ✔ | ✔ |
| Rechnungen erzeugen/versenden | ✖ | ✖ | ✔ | ✔ |
| CRUD | ✖ | ✖ | ✖ | ✔ |

## 6. Fehlercodes (Beispiele)

* RESERVATION\_CONFLICT – Zeitkollision
* AIRCRAFT\_GROUNDED – Aktion auf gegroundetem Flugzeug
* COUNTER\_MISMATCH – Startzähler passt nicht
* ROLE\_FORBIDDEN – Rolle verletzt

## 7. Konfigurierbarkeit

* Preislisten versioniert ab Datum
* Pflichtfelder je Klasse

## 8. Offene Punkte (optional, falls später gebraucht)

* DSGVO: Datenminimierung, Löschkonzept